UNISYS

A Series Communications Management System (COMS)

Migration Guide

Relative to Release
Level Mark 3.7.0

Priced Item

July 1987
Distribution Code SE
Printed in U S America
1185204.370

UNISYS

A Series Communications Management System (COMS)

Migration Guide

Copyright© 1987, Unisys Corporation
All Rights Reserved
Unisys is a trademark of Unisys Corporation

Relative to Release
Level Mark 3.7.0

Priced Item

July 1987
Distribution Code SE
Printed in U S America
1185204.370

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual living or otherwise, or that of any group or association, is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication may be forwarded using the Remarks form at the back of the manual; submitted on a Field Communication Form with the Class specified as 2 (System Software), the Type specified as 1 (F.T.R.), and the Product specified as the form number of this manual (1185204.370); or addressed directly to Unisys Corporation, Software Documentation, 19 Morgan, Irvine, California, 92718, U.S.A.

UNISYS

Publication Change Notice (PCN)

Date: May 28, 1989

Form—PCN number: 1185204-001

Title: A Series Communications Management System (COMS) Migration Guide (Relative to the Mark 3.8.0 System Software Release)

Description

This letter announces the availability of PCN-001 to the A Series Communications Management System (COMS) Migration Guide, form number 1185204, dated July 1987. This PCN includes revisions and additions relative to the Mark 3.8.0 System Software Release.

This PCN contains changes to the instructions for creating windows and using agendas when formatting.

Changes to the text are indicated by vertical black lines on the affected pages.

Replace These Pages	Add These Pages
iii	i
vii	42A
1	
39 through 41	
89 through 93	

Retain the PCN cover sheet as a record of changes made to the basic publication.

Copyright (c) 1989 Unisys Corporation
Unisys Corporation
All Rights Reserved

Printed in U S America

Page Status

Page	Issue
i	PCN-001
ii	Blank
iii through iv	PCN-001
v through vi	Original
vii through viii	PCN-001
1 through 2	PCN-001
3 through 38	Original
39 through 42A	PCN-001
42B	Blank
43	Original
44	Blank
45 through 85	Original
86	Blank
87 through 88	Original
89 through 93	PCN-001
94	Blank

ABOUT THIS GUIDE

The Communications Management System (COMS) is a Message Control System (MCS) designed for use with Unisys A Series systems and B 5900, B 6900, and B 7900 systems. The COMS product is a member of the InterPro (tm)* family of products.

PURPOSE

This guide provides information that is useful to personnel who are responsible for their site's migration from existing MCSs to COMS.

SCOPE

The guide details the strategies, tools, and methods of evaluation available for migration to COMS.

AUDIENCE

The guide is written for the system administrators and system programmers who are responsible for their site's migration.

PREREQUISITES

The users of the guide must be familiar with their site's existing MCSs and with COMS.

---

*InterPro is a trademark of Unisys Corporation.

HOW TO USE THIS DOCUMENT

This guide is divided into five sections. Read section 1 when you are planning or reviewing your strategy for migrating to COMS. Section 2 is important if you intend to operate a user-created or Unisys MCS under COMS.

Sections 3 and 4 detail different aspects of migrating from the Generalized Message Control System (GEMCOS) to COMS, and should be read if you are performing this migration.

Section 5 gives you information on how remote files are handled if you migrate to COMS, and compares this handling with that of the Command AND Edit (CANDE) system.

For convenience, the symbols that appear in this guide are used in the same manner as they are used in related documents. For example, brackets ([]) are used as part of GEMCOS declarations, and angle brackets (<>) denote metatokens and variable information.

Since the guide is addressed to users who are familiar with their site's existing MCSs and COMS, many of the terms, acronyms, and concepts in the guide that are integral to COMS, GEMCOS, or other Unisys MCSs are not included in the glossary or index.

Unless otherwise stated, all documents referenced in the guide are A Series documents.

ORGANIZATION

Sections

1 MIGRATING TO COMS

This section outlines the major benefits of using COMS, the migration strategies, and the factors to consider when selecting a migration path for your site.

2 MCS WINDOWS

This section describes the MCS window feature of COMS and provides information to help system programmers decide if any changes are needed to make a given MCS operate in a COMS environment.

3 FORMAT SUPPORT LIBRARY

One tool in the direct migration from GEMCOS to COMS, the Format Support Library processing item, is examined. The section covers the preparation, installation, and use of the Format Support Library to retain your GEMCOS input formatting, output formatting, form requests, and paging.

4 GEMCOS WINDOWS

Guidelines are presented for running GEMCOS in an MCS window, and various aspects of GEMCOS operation through COMS are considered.

5 REMOTE FILES

This section compares the handling of remote files through COMS with the handling of remote files through CANDE. Emphasis is placed on how COMS handles remote files.

Appendixes

A GEMCOS-TO-COMS FUNCTION AND FEATURE COMPARISON

This appendix compares GEMCOS and COMS functions and features.

B GEMCOS SOURCES FOR COMS ENTITIES AND ATTRIBUTES

This appendix gives the GEMCOS sources for COMS entities and attributes.

In addition, a glossary, a bibliography, and an index appear at the end of this guide.

RELATED PRODUCT INFORMATION

For more information about COMS, refer to the following documents:

Title	Form No.
Communications Management System (COMS) Capabilities Manual	1180494
Communications Management System (COMS) Configuration Guide formerly Communications Management System (COMS) Planning and Installation Manual	1185238
Communications Management System (COMS) Operations Guide formerly Communications Management System (COMS) Operator's Guide	1154523
Communications Management System (COMS) Programming Guide formerly Communications Management System (COMS) Programmer's Guide	1185303

CONTENTS

ABOUT THIS GUIDE. iii

1 MIGRATING TO COMS. 1
WHY COMS?. 1
Full-Featured COMS 1
COMS Windows and Dialogs 2
MIGRATING DIRECTLY OR USING A COMS MCS WINDOW. 4
MIGRATING DIRECTLY TO COMS 5
Migrating Directly from GEMCOS to COMS 5
Translating a GEMCOS TCL Control File. 6
GEMCOS Features in a COMS Environment. 7
GEMCOS Formatting Under COMS 7
OPERATING UNDER AN MCS WINDOW. 8
Accessing an MCS 8
Performance Under a COMS Window. 8
Using the GEMCOS Window. 8
MIGRATION ASSISTANCE 10

2 MCS WINDOWS. 11
GENERAL DESCRIPTION OF A COMS MCS WINDOW 13
MCS-CONTROLLED PSEUDO-STATIONS 14
NAMING CONVENTION FOR PSEUDO-STATIONS. 15
REQUIREMENTS FOR AN MCS TO FUNCTION THROUGH COMS 16
ENTERING CONTROL COMMANDS. 17
HANDLING OF DCWRITEs 18
DCWRITEs Handled by the MCP. 18
DCWRITEs Interpreted by COMS 19
Result Messages Given by COMS. 23
DCWRITEs Not Allowed by COMS 24
FORMATS OF DCWRITE MESSAGES AND RESULT MESSAGES. 26
DCWRITE Message Format 26
Result Message Formats 28

3 FORMAT SUPPORT LIBRARY 31
WHY YOU SHOULD USE THE FORMAT SUPPORT LIBRARY. 32
USING THE FORMAT SUPPORT LIBRARY WITH SDF. 33
GEMCOS FORMATTING RETAINED IN COMS 34
Formatting Input and Output Messages 34
Making a Forms Request 34
Dividing Messages into Pages 34
User-Created Formats 35
USING THE FORMAT SUPPORT LIBRARY 36
Using the Format Support Library in Agendas. 36
Formatting Performance Expectations. 36
Logical Flow for Input and Output Formatting 36
Logical Flow for Forms Request 37
Logical Flow for Paging. 37
CREATING THE COMS ENVIRONMENT. 39
Creating COMS Windows for Systems in the GEMCOS TCL. . . 39

Defining Trancodes 40
Defining Device Types. 40
Defining Processing-Item Libraries 40
Defining Processing Items. 41
Creating Processing-Item Lists 41
Using Agendas When Formatting. 42
Updating Formats 42
Amending Application Programs. 42
Obtaining Monitor Output 42
WHAT THE GEMCOS TCL FILE MUST CONTAIN. 43

4 GEMCOS WINDOWS 45
GENERAL DESCRIPTION OF A GEMCOS WINDOW 46
ADVANTAGES OF RUNNING GEMCOS THROUGH COMS. 47
CHANGING STATION NAMES IN GEMCOS TCL 48
ENTERING CONTROL AND ODT COMMANDS. 50
GEMCOS FEATURE LIMITATIONS WITH THE CP2000 52
Limitations Due to Loss of Toggles and Tallies 52
Other Limitations. 53
RELATIVE GEMCOS FUNCTIONAL BEHAVIOR WITH CP2000 AND NSP. . 54

5 REMOTE FILES 55
WHAT IS A REMOTE FILE? 55
SITUATIONS FOR USING REMOTE-FILE WINDOWS 55
COMPARISON OF COMS AND CANDE REMOTE-FILE HANDLING. 55
Dynamic and Declared Remote Files. 56
Multistation Remote Files. 57
Input Remote Files 57
Defining and Executing Declared Remote-File Programs . . 57
Executing a Remote-File Program by a RUN Command 58
Closing a Remote File. 58
Tanking. 60
Handling of Time Limit on a Write Operation. 60
Direct I/O 60

A GEMCOS-TO-COMS FUNCTION AND FEATURE COMPARISON 61

B GEMCOS SOURCES FOR COMS ENTITIES AND ATTRIBUTES. 73

GLOSSARY. 77

BIBLIOGRAPHY. 87

INDEX 89

1 MIGRATING TO COMS

This section outlines the benefits of the full-featured Communications Management System (COMS), the available routes and tools for migration to COMS, and the factors that influence migration.

WHY COMS?

COMS is part of the Unisys InterPro (tm) family of products. It serves as the Message Control System (MCS) and supports processing on the Unisys A Series systems and B 5900, B 6900, and B 7900 Series systems.

COMS allows you to select only the features you need for your processing. You can choose features such as a menu-driven user interface provided by the Menu-Assisted Resource Control (MARC), transaction processing, and real-time configuration changes.

COMS windows allow you to operate a number of program environments simultaneously at a given station. Without logging off, you can switch from window to window and retain the states of your window dialogs.

The close integration of COMS with the Master Control Program (MCP) guarantees high performance, especially for the direct window interface.

The available COMS processing items and windows are important factors in your migration plan.

Full-Featured COMS

The full-featured version of COMS offers the following:

- Eight dialogs each on the Menu-Assisted Resource Control (MARC) window, the Command AND Edit (CANDE) window, and the Generalized Message Control System (GEMCOS) window.
- Preprocessing and postprocessing of messages.
- Routing by transaction code (trancode).
- User-defined usercodes and station restrictions.
- Dynamic control of user-created interface programs.

- Security options, such as security by trancode.
- Batch processing.
- Synchronized recovery.
- GEMCOS-to-COMS migration aids.

COMS Windows and Dialogs

COMS windows and dialogs are basic to most migration strategies. Four windows are defined by COMS at system initialization:

- MARC window
- COMS Utility window
- CANDE window
- GEMCOS window

The MARC window handles network control, datacomm errors, and COMS termination. This window is a menu-driven interface to system resources that can be selected for an installation. The MARC window is similar to CANDE in the way it runs and controls application programs.

The COMS Utility window is accessible only from a control station, the Operator Display Terminal (ODT), or through a control-capable usercode. Through the COMS Utility window, you can define and maintain the COMS control file.

The CANDE and GEMCOS windows provide access to CANDE and GEMCOS, respectively.

There are three types of COMS windows:

- Direct windows
- MCS windows
- Remote-file windows

Direct windows are used to perform transaction-oriented processing that provides data base synchronized recovery. MCS windows access message control systems other than COMS. Remote-file windows access remote-file programs.

The MARC and Utility windows are direct windows. The CANDE and GEMCOS windows are MCS windows. Both direct and MCS windows can be declared through the COMS Utility. Remote-file windows can be declared through the COMS Utility, or dynamically created by executing a program that opens a remote file.

Access to a program environment through a window is called a dialog. COMS allows multiple and simultaneous dialogs with the same window during the same session. Each dialog is treated as an independent session with a program environment.

MIGRATING DIRECTLY OR USING A COMS MCS WINDOW

When you migrate directly to COMS, you replace your current Message Control System (MCS) with COMS. If you decide to delay replacement of your MCS with COMS, you can establish a COMS MCS window in which to operate your MCS.

Your current MCS, the size and complexity of your current applications, available expertise, hardware, and the features and functions offered by COMS all influence your choice of a migration strategy.

MIGRATING DIRECTLY TO COMS

The direct approach consists of performing all required tasks to replace your current MCS or MCSs with COMS at one time. Depending on your MCS or MCSs, direct migration can mean simultaneously modifying your data base, changing your application programs, and creating COMS entities.

For information on modifying your data base and creating COMS entities, refer to the COMS Configuration Guide. The COMS Programming Guide contains information on how to write application programs for transaction processing that run under COMS.

Direct migration might involve replacing a user-created MCS or MCSs with COMS. For example, user-created dial-out MCSs can be replaced with COMS-supplied dial-out facilities. Remote printing MCSs can be replaced with Unisys Remote Print Systems (ReprintS). (Under ReprintS, the remote destination must be a datacomm station controlled by MARC and COMS.)

Direct migration from GEMCOS to COMS involves the specific steps and considerations outlined in the paragraphs that follow. The Format Support Library (FS-LIB) described in the "Format Support Library" section can assist in your direct migration strategy.

See also
    Format Support Library. 31

Migrating Directly from GEMCOS to COMS

If you currently use GEMCOS, you can migrate directly to COMS if you alter your application programs and COMS to emulate or accommodate the features of GEMCOS.

The following five steps outline the basic strategy of a direct migration from GEMCOS to COMS.

1. Install GEMCOS on a Mark 3.5-or-later Master Control Program (MCP).

2. Install COMS and MARC.

3. Change the data base restart data set to support COMS recovery.

4. Convert your GEMCOS application programs into programs that use COMS programming conventions.

5. Translate the network description in the GEMCOS Transaction Control Language (TCL) control file into a network description that can be understood and used by COMS.

COBOL68 programs do not need to be converted to COBOL74. However, COBOL74 has language support for the COMS interface. With COBOL68 you must program the interface using LIBRARY calls.

Translating a GEMCOS TCL Control File

For the direct migration to be successful, the network description specified in the TCL control file needs to be translated into a network description that can be understood and used by COMS. GEMCOS entities (categories of information in the control file) need to be translated into COMS entities.

Most GEMCOS entities have a direct COMS counterpart. For example, the GEMCOS entity called a SYSTEM is the counterpart of the COMS WINDOW entity. Table 1-1 lists several GEMCOS entities and their COMS counterparts. The "GEMCOS Sources for COMS Entities and Attributes" appendix gives the GEMCOS counterpart for each COMS entity.

See also
GEMCOS Sources for COMS Entities and Attributes 73

Table 1-1. Comparison of GEMCOS and COMS Entities

GEMCOS	COMS
SYSTEM	WINDOW
PROGRAM	PROGRAM
INPUTQUEUE	AGENDA
STATION	STATION
DEVICE	DEVICE_TYPE

GEMCOS Features in a COMS Environment

Before beginning a direct migration, read the "GEMCOS-to-COMS Function and Feature Comparison" appendix to see if the GEMCOS features you need are supported under COMS.

If you need another feature, consider creating your own processing item to support the feature.

See also
GEMCOS-to-COMS Function and Feature Comparison. 61

GEMCOS Formatting Under COMS

The Format Support Library available with COMS provides GEMCOS formatting and paging features in the COMS environment. The Screen Design Facility (SDF), another InterPro (tm) product, also can be used to provide formatting for devices compatible with ET 1100 terminals.

OPERATING UNDER AN MCS WINDOW

Under COMS, a window provides access to a program or system. MCS windows give you access to Message Control Systems (MCSs). A site can have a maximum of 1023 windows. Only four windows (MARC, COMS Utility, CANDE, and GEMCOS) are defined at system initialization. You can define additional windows any time. You could define a window for each message control system supported at your site. Refer to the "MCS Windows" section for more information on MCS windows.

See also
MCS Windows 11

Accessing an MCS

When you access an MCS through a COMS window, the MCS and its application programs do not have to be modified. (This is a major difference from direct migration, in which the programs must be modified.) For example, CANDE and programs running under CANDE need no modifications to operate under a COMS window. However, as shown below, using an MCS window creates an extra processing layer.

Performance Under a COMS Window

There are no performance benefits when you run any MCS under a COMS window.

Using the GEMCOS Window

The Kernel version of COMS allows a station to have one dialog of the GEMCOS window. The Kernel version does not offer the conveniences and benefits of the Entry and full-featured versions of COMS. The Entry and full-featured versions allow multiple and simultaneous dialogs with GEMCOS at one station and give current GEMCOS users the windows and menu-driven features provided by COMS. However, the GEMCOS window should be used only as a short-term strategy in the migration to COMS.

Figure 1-1 shows what happens when you run GEMCOS under an MCS window. The extra layer, COMS itself, is used when any MCS operates under a window. This extra layer can hinder performance.

GEMCOS Alone:

Application Program <----> GEMCOS <----> Hardware

GEMCOS Under COMS:

Application Program <----> GEMCOS <----> COMS <----> Hardware

Figure 1-1. Adding a Software Layer by Running GEMCOS Under a COMS Window

The GEMCOS window, in most situations, should be used as an interim tool before you begin converting your GEMCOS application programs. However, CP2000 users must use the GEMCOS window; GEMCOS cannot directly access CP2000 stations. For more details about GEMCOS windows, read the "GEMCOS Windows" section.

See also
GEMCOS Windows. 45

MIGRATION ASSISTANCE

If you have a migration solution that is not covered in this document or if you have created a migration tool that you would like to share with others, please send a detailed description to your local Unisys technical representative.

If you have a problem that requires immediate attention, contact your local Unisys technical representative.

2 MCS WINDOWS

This section describes the characteristics of a Communications Management System (COMS) Message Control System (MCS) window and indicates how it works. The section describes how COMS handles various DCWRITEs with respect to an MCS window, and presents DCWRITE-message and result-message formats. DCWRITEs are intrinsic functions that cause messages using these functions to be passed to the datacomm controller (DCC) for action.

The information in this section pertains to a COMS MCS window in general, that is, a COMS MCS window that is used for any given Message Control System (MCS). For information about running the Generalized Message Control System (GEMCOS) in a COMS MCS window, refer to the "GEMCOS Windows" section.

See also
GEMCOS Windows. 45

The following topics are presented in this section:

- General description of a COMS MCS window
- MCS-controlled pseudo-stations
- Naming convention for pseudo-stations
- Requirements for an MCS to function through COMS
- Entering control commands
- Handling of DCWRITEs
- DCWRITE-message and result-message formats

For additional information about pseudo-stations and DCWRITEs, refer to the DCALGOL Programming Reference Manual.

Unless indicated otherwise, the information in this section applies to an MCS-controlled station or pseudo-station operating either with the CP2000 front-end processor through COMS or with the Network Support Processor (NSP) through COMS.

Note that a given MCS operating with the NSP through COMS may have more capabilities than the same MCS operating with the CP2000 through COMS.

Note also that an installation that includes both the CP2000 and the NSP can have a mixed network in which stations and pseudo-stations controlled by an MCS can operate individually. These stations and pseudo-stations can function in any or all of the following ways:

- With the CP2000 through COMS
- With the NSP through COMS
- Directly under the NSP

Note further that the LSCANDE virtual-terminal protocol must be used in order for a terminal to operate with the CP2000 through COMS.

GENERAL DESCRIPTION OF A COMS MCS WINDOW

A COMS MCS window is a feature that enables you to have a number of independent sessions (dialogs) with an MCS simultaneously at a single station, if that MCS is installed on the network. Examples of MCSs are the Command AND Edit (CANDE) system, GEMCOS, and user-provided MCSs.

The Kernel version of COMS allows a station to have up to two dialogs of the CANDE window and one dialog of the GEMCOS window at the same time.

The Entry version and full-featured version of COMS provide a means to define MCS windows through the COMS Utility program, as described in the COMS Configuration Guide. The COMS Utility can be used to define a number of MCS windows for one or more kinds of MCSs.

Once an MCS window is defined through the COMS Utility program for an installed MCS, you can have a number of simultaneous dialogs with the MCS at your station. The window can be defined through the COMS Utility to allow a maximum of eight simultaneous dialogs with the MCS at any station. You can then perform various activities associated with that MCS at the same time on one physical terminal.

In general, the windowing capabilities of COMS allow you to perform a variety of activities simultaneously on the terminal. These activities can be associated with one or any number of MCSs and/or other systems that have been installed and window-defined.

For information about controlling window dialogs at a terminal, refer to the COMS Operations Guide.

MCS-CONTROLLED PSEUDO-STATIONS

When you open a dialog of a COMS MCS window at your station, the dialog is established through a Master Control Program (MCP) feature called a "pseudo-station." A pseudo-station is a virtual station that can be attached to and controlled by an MCS in much the same way as a real station. Unlike a real station, however, a pseudo-station is not declared in Network Definition Language II (NDLII), has no line assigned, and needs no corresponding physical terminal on the local host.

For every MCS window dialog that you open, COMS allocates a pseudo-station and transfers it to the MCS in question. Each DCWRITE that the MCS performs on the pseudo-station is handled in one of the following ways:

- Handled entirely by the MCP. The DCWRITE works as it would for a real station.
- Handled primarily by the MCP, with some interpretation by COMS.
- Interpreted by COMS.
- Rejected because it is not allowed on a pseudo-station.

The pseudo-station feature enables COMS to implement MCS windows. This feature does not apply to COMS direct windows or remote-file windows. When information is input through the MCS window dialogs at a given real station, the resulting output to the user is handled as if the response were to a number of real stations. In fact, only one terminal is involved.

In a COMS environment, the MCS or MCSs handling the input/output never come in contact with the name corresponding to the real station. Rather, the MCS or MCSs recognize the name of the pseudo-station corresponding to each dialog.

You can recognize output associated with a particular MCS window dialog by means of the pseudo-station name identifying that output.

NAMING CONVENTION FOR PSEUDO-STATIONS

Pseudo-station names are assigned automatically by COMS. When your station is on the local host, the naming convention for pseudo-stations is as follows:

<NDLII station name>/<MCS window name>/<dialog number>

Example:

ABC/CANDE/2

When your station has been transferred to another host through the Burroughs Network Architecture (BNA), the naming convention for pseudo-stations is as follows:

<system host ID>/<NDLII station name>/<MCS window name>/<dialog number>

Example:

A9A/ABC/CANDE/2

REQUIREMENTS FOR AN MCS TO FUNCTION THROUGH COMS

In general, an MCS that presently works with real stations on lines needs no modification to function in a COMS MCS window if the MCS satisfies all the following requirements:

1. The MCS does not identify its stations using the NSP Line Station (NLS) number format, shown below:

   <relative NSP number>:<line number>:<station number>

   Instead, the MCS identifies its stations using the logical station number (LSN).

2. The MCS does not have to reconfigure the network and handle switched lines as part of the application.

3. The MCS does not communicate with the datacomm subsystem through toggles and tallies.

ENTERING CONTROL COMMANDS

The following information is provided to help a station user communicating through a COMS MCS window enter control commands directed to COMS or the given MCS.

If you enter a control command prefixed by one question mark (?), COMS always responds to the command, even if the command syntax is identical to that of a control command associated with the given MCS. If your entered control command prefixed by one question mark is not a COMS command, it is sent to the MCS.

If you enter a control command associated with the given MCS prefixed by two question marks (??), the command is always sent to the MCS, even if the command syntax is identical to that of a COMS control command.

HANDLING OF DCWRITEs

DCWRITEs are DCALGOL intrinsic functions that cause specified messages constructed to use these functions to be passed to the Data Communications Controller (DCC) for action. The action that the DCC takes on a message depends on the type and variant fields of the message.

Information about the handling of DCWRITEs for a pseudo-station in a COMS MCS window is presented in the paragraphs that follow. The information is presented in tables under the following categories:

- DCWRITEs handled by the MCP
- DCWRITEs interpreted by COMS
- Result messages given by COMS
- DCWRITEs not allowed by COMS

DCWRITEs Handled by the MCP

Table 2-1 lists the type numbers and corresponding functions for DCWRITEs that either are handled entirely by the MCP or primarily by the MCP for a pseudo-station in a COMS MCS window.

COMS is notified of the DCWRITEs that are handled primarily by the MCP, and COMS takes some action on these DCWRITEs. In Table 2-1, these DCWRITEs are identified by the parenthetical note "(some COMS action taken)." The action that COMS takes on each of these DCWRITEs is indicated in Table 2-2.

Table 2-1. DCWRITEs Handled by the MCP

DCWRITE Type Number	DCWRITE Function
00	Initialize primary queue
01	Attach station (some COMS action taken)
02	Interrogate MCS
03	Communicate among MCSs
04	Interrogate station environment
05	Attach schedule station
32	Change current queue (some COMS action taken)
42	Detach station
45	Transfer station control
64	Assign station to file (some COMS action taken)
65	Write to object job
66	Cause station break
67	Add station to file (some COMS action taken)
69	Subtract station from file
129	Exchange clusters
131	Update line attributes

DCWRITEs Interpreted by COMS

Table 2-2 lists the type numbers and corresponding functions for DCWRITEs that are interpreted by COMS for a pseudo-station in a COMS MCS window. The action that COMS takes on each of these DCWRITES is indicated in the table.

Note that the expression "giving dummy good result to MCS" in the table means that COMS is notifying the MCS that the function associated with the particular DCWRITE has been accomplished. The expression does not mean that COMS performs the function.

Table 2-2. DCWRITEs Interpreted by COMS (Page 1 of 3)

DCWRITE Type Number	DCWRITE Function	COMS Action
01	Attach station	Determine whether or not a good result is required by the MCS.
32	Change current queue	Determine whether or not a good result is required by the MCS.
33	Write	Send text; return the result to the MCS if wanted.
34	Read-once only	No action.
35	Enable input	Indicate that the input has been enabled by giving a dummy good result to the MCS.
36	Disable input	Indicate that the input has been disabled by giving a dummy good result to the MCS.
37	Make station ready/not ready	Resume or suspend window dialog. Make pseudo-station ready or not ready. Give results to the MCS.
38	Set application number	Indicate that the application number has been set by giving a dummy good result to the MCS.

Table 2-2. DCWRITEs Interpreted by COMS (Page 2 of 3)

DCWRITE Type Number	DCWRITE Function	COMS Action
39	Set characters	Indicate that the characters have been set by giving a dummy good result to the MCS.
40	Set transmission number	Indicate that the transmission number has been set by giving a dummy good result to the MCS.
41	Recall message for the following four situations:	
	a. COMS is recalling	Give a result to the MCS indicating that no message has been recalled.
	b. MCS wants only first message	Give a result to the MCS indicating that no message has been recalled.
	c. Object job output	Recall the first message. Give it to the MCS.
	d. Else	Recall all messages. Give them to the MCS.
43	Set/reset LOGICALACK (logical acknowledgement)	Indicate that LOGICALACK has been set or reset by giving a dummy good result to the MCS.
44	Acknowledge	Indicate that acknowledgement has been made by giving a dummy good result to the MCS.
46	Write and return	Send text; return the result if wanted.
48	Request null station	Issue a null request (for NSP) or issue minor synchronization (for CP2000).

Table 2-2. DCWRITEs Interpreted by COMS (Page 3 of 3)

DCWRITE Type Number	DCWRITE Function	COMS Action
64	Assign station to file	Send text out if it applies to the message.
67	Add station to file	Send text out if it applies to the message.
68	Change terminal attributes	No action.
96	Make line ready	Indicate that the line has been made ready by giving a dummy good result to the MCS.
97	Make line not ready	Indicate that the line has been made not ready by giving a dummy good result to the MCS.
103	Set/reset line toggles/tallies	No action.
105	Force line not ready	No action.

Result Messages Given by COMS

Table 2-3 lists the values for and descriptions of DCWRITE result messages that COMS gives to the MCS for a pseudo-station in a COMS MCS window. An explanation of how COMS handles the result for each of these result messages is indicated in the table.

Table 2-3. DCWRITE Result Messages Given by COMS

Result Message Value	Message Description	Explanation
00	Good input	Message with text is input.
01	Station event	Only the control message variant of the input event type is given to the MCS.
02	File open	Not affected by COMS.
03	Object job output	Not affected by COMS.
04	File close	Not affected by COMS.
05	Good results	Good results are generated for all DCWRITEs that give 05 for a real station.
06	Recalled message	The message is recalled.
14	Station detached	The window is closed (that is, the pseudo-station with the specified Logical Station Number (LSN) is deallocated).
15	Interrogate station environment result	Pieces of information relating to the pseudo-station are returned.
16	Transfer station result	Not affected by COMS.
18	ODT-to-station result	Not affected by COMS.
99	Error result	An error result is generated only for the break case--that is, only for a break from a terminal device or printer.

DCWRITEs Not Allowed by COMS

Table 2-4 lists the type numbers and corresponding functions for DCWRITEs that are not allowed for a pseudo-station in a COMS MCS window.

Table 2-4. DCWRITEs Not Allowed by COMS

DCWRITE Type Number	DCWRITE Function
49	Set or reset sequence mode
53	Write to transferred station
55	Send result message to the MCS
56	Set pseudo-station attributes
98	Dial out
99	Disconnect
100	Answer the phone
101	Interrogate switched status
102	Set or reset autoanswer
128	Swap lines
130	Move, add, or subtract station

If an attempt is made to execute a DCWRITE that is not allowed on a pseudo-station in a COMS MCS window, one of the error messages listed in Table 2-5 is returned on the call.

Table 2-5. Error Messages

Error Message Value	Description
188	That DCWRITE is not allowed on a pseudo-station.
190	The pseudo-station already has a fully participating MCS. Therefore, the requested type of transfer-station DCWRITE cannot be granted.
192	That DCWRITE is allowed only by a fully participating MCS.

For a discussion of "full participation," refer to the DCALGOL Programming Reference Manual.

FORMATS OF DCWRITE MESSAGES AND RESULT MESSAGES

The formats of DCWRITE messages and result messages associated with particular DCWRITEs are presented in the paragraphs that follow. These formats are presented to show the way COMS handles fields in the header and in other words of these messages, such as the text.

DCWRITE Message Format

The format of DCWRITE messages used in connection with DCWRITE 33 (WRITE) and DCWRITE 46 (WRITE and RETURN) is presented in Table 2-6. Fields in this message format are identified and described in the table. The table indicates which fields are handled by COMS, and specifies how some of these fields are handled. The table also indicates fields that are not handled by COMS. Brief explanations of the fields handled by COMS follow the table.

Additional information about the fields described in Table 2-6 is presented in the DCALGOL Programming Reference Manual.

Table 2-6. DCWRITE Message Format

Word Field	Field Description	How Field is Affected by COMS
MSG[0].[47:08]	Type	Handled
MSG[0].[39:16]	Carriage control	Forwarded to the network
MSG[0].[23:24]	LSN	Mapped onto real station
MSG[1].[47:01]	(Not used)	
MSG[1].[46:07]	Priority	Not handled
MSG[1].[39:08]	Toggles	Not handled
MSG[1].[31:32]	(Not used)	
MSG[2].[47:08]	Retry count	Not handled
MSG[2].[39:16]	Text size	Handled
MSG[2].[23:24]	(Not used)	
MSG[3].[47:24]	(Not used)	
MSG[3].[23:24]	Tallies	Not handled
MSG[4].[47:24]	Message number	Saved for result message
MSG[4].[23:24]	(Not used)	
MSG[5]	(Not used)	
MSG[6]-MSG[N]	Text	Handled

The type field (MSG[0].[47:08]) contains a value that tells the DCWRITE function which operation to perform.

The carriage control field (MSG[0].[39:16]) provides information about advancement of position in the output device.

The LSN field (MSG[0].[23:24]) designates the pseudo-station to be affected by the DCWRITE function.

The text size field (MSG[2].[39:16]) specifies the number of bytes of text in the message minus the number of bytes in the message header.

The message number field (MSG[4].[47:24]) contains a value that identifies the message to the MCS.

The text field (MSG[6]-MSG[N]) contains the text of the message. This field is left justified and starts in word 6 of the message.

Result Message Formats

Table 2-7 presents the format of result messages returned for DCWRITE 33 (WRITE) and DCWRITE 46 (WRITE and RETURN).

Table 2-8 presents the format of result messages returned for DCWRITE 41 (RECALL message).

In these tables, fields in each of the result message formats are identified and described, and the returned information is shown. Brief explanations of fields for which information is returned follow the tables.

Additional information about the fields described in Tables 2-7 and 2-8 is presented in the DCALGOL Programming Reference Manual.

Table 2-7. Format of Result Messages Returned for DCWRITEs 33 (WRITE) and 46 (WRITE and RETURN)

Word Field	Field Description	Returned Information
MSG[0].[47:08]	Class	05
MSG[0].[39:16]	Variant	00
MSG[0].[23:24]	LSN	Pseudo-LSN
MSG[1]		00
MSG[2]		00
MSG[3]		00
MSG[4].[47:24]	Message number	Value in original DCWRITE
MSG[4].[23:24]	Original DCWRITE	Original DCWRITE (33 or 46)
MSG[5]		00
MSG[6]-MSG[N]	Text	Not returned

Table 2-8. Format of Result Messages Returned for DCWRITE 41 (RECALL Message)

Word Field	Field Description	Returned Information
MSG[0].[47:08]	Class	06
MSG[0].[39:16]	Variant	00 or 01
MSG[0].[23:24]	LSN	Pseudo-LSN
MSG[1]		00
MSG[2]		00
MSG[3]		00
MSG[4].[47:24]	Message number	Value in original DCWRITE
MSG[4].[23:24]		00
MSG[5]		00
MSG[6]-MSG[N]	Text	Not returned

The class field (MSG[0].[47:08]) contains a value that determines the way in which the remainder of the message is interpreted.

The variant field (MSG[0].[39:16]) indicates qualification, variations, or additional information concerning the message type.

The LSN field (MSG[0].[23:24]) designates the pseudo-station destination for the result message.

The message number field (MSG[4].[47:24]) contains a value that identifies the original DCWRITE to the MCS.

3 FORMAT SUPPORT LIBRARY

With Communications Management System (COMS) direct windows, you can process messages before they are received by programs on input (preprocessing) or before they are sent to a device on output (postprocessing). Processing is performed by processing items. The processing items must be defined in the COMS configuration file via the COMS Utility.

Processing items are grouped into processing-item lists. Processing-item lists are placed in agendas to determine which items to apply to messages that invoke the agendas on input or output. The agendas are also defined to route the messages to destination programs for the agenda.

The ability to direct preprocessing and postprocessing of messages makes a processing item an excellent tool to maintain Generalized Message Control System (GEMCOS) formatting and paging in the COMS environment. COMS provides a Format Support Library (FS-LIB) with an ACTUALNAME of TCL_FORMATTER, to maintain GEMCOS input formatting, output formatting, forms request, paging, and library formats.

The Format Support Library should be included in any COMS agenda that needs to perform GEMCOS message formatting. This processing item can be used in a direct migration approach and with the Screen Design Facility (SDF). It cannot be used with any Message Control System (MCS) window.

This section covers the following topics:

- When and how to use the processing item.
- GEMCOS formatting retained under COMS through the processing item.
- The logical flow of the processing item.
- Performance when using the processing item.
- The needed COMS environment.
- How to prepare the GEMCOS Transaction Control Language (TCL) file to work in the COMS environment.

WHY YOU SHOULD USE THE FORMAT SUPPORT LIBRARY

The Format Support Library is designed to be used in the direct migration from GEMCOS to COMS. It emulates GEMCOS formatting features in the COMS environment. The library allows you to maintain the formatting specified in the GEMCOS TCL file while running under COMS. If another feature or function is needed, you can create your own processing item and include it in the appropriate agenda or agendas.

As described below, the Format Support Library can be used with the Screen Design Facility (SDF). You can use SDF instead of the processing item, or you can use SDF with the processing item. You cannot use the Format Support Library when running programs under any MCS window.

USING THE FORMAT SUPPORT LIBRARY WITH SDF

SDF is an InterPro (tm) product designed to create and maintain forms. SDF generates processing items that are used by COMS for formatting.

The Format Support Library and the SDF processing items can be used at the same time. The SDF processing items should be placed after the Format Support Library in the agenda's processing-item list for both input and output messages.

If the Format Support Library does any formatting, it sets the first word of the program's conversation area to all 1s. SDF processing items examine the first word of the program's Conversation area on both input and output. If the word has all bits set to 1, SDF knows the Format Support Library has done the formatting; SDF does not process the data. If the word does not have all bits set to 1, processing continues and SDF formatting is performed.

GEMCOS FORMATTING RETAINED IN COMS

Formatting defines how a message is handled prior to its delivery to a program (input messages) or an output device (output messages). The Format Support Library emulates GEMCOS formatting by performing the following functions:

- Formatting input and output messages
- Handling forms request
- Handling paging
- Supporting user-created format libraries

Formatting Input and Output Messages

Stations are grouped by device class. By identifying the device class and either the message key or message ID, the Format Support Library applies the proper format to a message.

Making a Forms Request

A forms request is a request for a blank form. When the request is processed, the terminal displays only prompts and format-supplied defaults; no previously entered values are shown. To make a forms request, a station transmits an output message ID with no data and no special terminating character.

Dividing Messages into Pages

Paging is dividing a message into distinct segments. This is usually done when

- The message is too long or crowded for the entire message to be displayed on the terminal at one time.
- The programmer wants to make specific distinctions between parts of the message.

A message can be divided into two or more pages. A given message can be paged at one device type and not paged at another device type. Both input and output messages can be paged.

When a station enters paging mode, COMS examines the first character of transmitted data to determine what action to take. The same characters that control the GEMCOS paging dialogue also control the COMS Format Support Library paging dialogue.

User-Created Formats

The GEMCOS Format Library interface allows users to create message formats using ALGOL, COBOL68, or COBOL74. These formats are maintained in addition to the standard TCL formats. All the user-created formats must be maintained in one format library. The title of the format library must be specified in the global section of the TCL file. If no title is specified, the default title is GEMCOS/FORMATLIBRARY.

USING THE FORMAT SUPPORT LIBRARY

All GEMCOS formats for input, output, forms requests, paging, and library are retained in COMS through the use of the Format Support Library.

The following paragraphs discuss various aspects of the Format Support Library.

Using the Format Support Library in Agendas

The Format Support Library must be included in an agenda to perform input or output formatting, a forms request, or paging. If SDF processing items are also used, the SDF processing items must be included after the Format Support Library.

Formatting Performance Expectations

Using the Format Support Library could improve your system's formatting performance because the library runs as part of the COMS multiprogramming environment, whereas GEMCOS uses single-streamed formatting. At minimum, your system should format as efficiently using the Format Support Library as it does using GEMCOS.

Logical Flow for Input and Output Formatting

When input or output formatting is required, the Format Support Library follows the basic logical flow shown below:

1. Determines the window where the message was entered or is being sent and locates the appropriate TCL system.

2. Gets the device type for the station where the input message was entered or where the output message is to be sent.

3. Searches for the proper format using the system index, the device type, and either the format ID or message key.

4. Establishes parameters and calls the formatter routine to format the message.

5. Exits.

Logical Flow for Forms Request

When a forms request is entered, the Format Support Library varies the basic logical flow. Steps 1, 2, and 5 of a forms request are the same steps used for input and output formatting. Steps 3 and 4 are different.

1. Determines the window where the message was entered or is being sent, and locates the appropriate TCL system.

2. Gets the device type for the station where the input message was entered or where the output message will be sent.

3. For input messages with lengths less than or equal to six characters, searches among all message IDs to see if the user is requesting a form.

4. For a forms requestor, sends the form to the requesting station.

5. Exits.

Logical Flow for Paging

When a user application program sends a message that requires paging, the basic logical flow followed by input and output formatting is initiated. At step 4, the Format Support Library calls paging routines to process the message. Steps 4, 5, and 6 are directed by the paging routines.

1. Determines the window where the message was entered or is being sent, and locates the appropriate TCL system.

2. Gets the device type of the station where the input message was entered or where the output message will be sent.

3. Searches for the proper format using the system index, the device type, and either the format ID or message key.

4. For paged formats, calls paging routines to send the first screen to the device and to prepare for paging dialogue.

5. For update paging, when the user sends the paging complete character (X), sends the entire formatted message to COMS as a new input.

6. Exits.

Formatting is then handled by paging routines until the paging mode is exited.

With the Format Support Library, there is no restriction on the number of stations that can use paging at one time even if GEMCOS control paging is specified in the TCL.

CREATING THE COMS ENVIRONMENT

To produce the correct formatting, the Format Support Library requires the following COMS entities:

1. Windows
2. Trancodes
3. Device types
4. Libraries
5. Processing items
6. Processing-item lists
7. Agendas

Create these entities through the COMS Utility before using the Format Support Library. (See the "GEMCOS Sources for COMS Entities and Attributes" appendix for a listing of GEMCOS and COMS counterparts.)

Because of the importance of the COMS entities listed above to the Format Support Library, information about defining them to COMS is presented in the paragraphs that follow. The COMS Utility command for creating each of these entities is included. Also presented is information about updating formats, amending application programs to use the COMS direct interface, and obtaining monitor output.

See also
GEMCOS Sources for COMS Entities and Attributes 73

Creating COMS Windows for Systems in the GEMCOS TCL

For every system in the GEMCOS TCL for which you want to use formatting in COMS, create a window with the same name as the system. In the TCL, at least one program must be declared in a system. All message keys for a system must be declared under that program.

The COMS Utility command for creating a window is

CREATE WINDOW <window name>

Defining Trancodes

The COMS Utility command for defining a trancode to COMS is shown below. Ensure that the trancode (message key) from the TCL matches the input in the command.

```
CREATE TRANCODE <trancode name> OF <window name>
AGENDA = <agenda name>
FUNCTION = <integer>;
```

Note that all trancodes must start in position one of the input. Also note that the Format Support Library discontinues (DS) the task if a conversation area is not allocated in the header of the program.

Defining Device Types

The COMS Utility command for defining a device type to COMS is shown below. Ensure that the device declared in the TCL matches the input in the command.

```
CREATE DEVICE_TYPE = <device_type name>;
CREATE STATION <station name>
   DEVICE_TYPE = <device_type name>;
```

Defining Processing-Item Libraries

In COMS, a library contains a set of procedures called processing items. These items can be called by agendas to process messages before or after input or output. A library name must be assigned to each processing item defined in the configuration file.

The COMS Utility command for defining a processing-item library to COMS is

```
CREATE LIBRARY <library name>;
```

You can have multiple copies of the Format Support Library to process different TCL files. Each copy must be a separate copy of the code file, and each can have one GEMCOS FORMATLIBRARY.

Defining Processing Items

The processing item ACTUALNAME must be TCL_FORMATTER. It must be associated with a previously defined processing-item library.

The COMS Utility command for defining a processing item to COMS is

```
CREATE PROCESSING_ITEM <processing item name>
   ACTUALNAME = TCL_FORMATTER,
   LIBRARY = <library name>;
```

Creating Processing-Item Lists

The processing-item list contains the processing items to be called by the agenda before it processes the message. The order in which the processing items appear in the list is the order in which they are called. If the Format Support Library is used in conjunction with SDF, the SDF processing items must follow the Format Support Library.

Include the Format Support Library in an agenda's processing-item list when any input or output formatting, forms request, or paging is required. Include it in the default agenda for windows that might use paging. When COMS encounters invalid trancodes, such as paging dialogue characters, the default agenda is invoked.

The COMS Utility command for creating a processing-item list is

```
CREATE PROCESSING_ITEM_LIST
   <processing_item_list name> =
      <processing_item name1, name2, name3,...>;
```

Using Agendas When Formatting

A COMS agenda processes and routes messages on input, output, or both, for direct windows. To use an agenda on input, the agenda must have an assigned destination program. Form requests and paging dialogue messages, however, do not go to the destination program.

To use update paging with a particular window, the window must have an additional agenda created with INPUT_ROUTER as its destination. This agenda must have the same name as the window and the GEMCOS TCL system.

The COMS Utility command for defining an agenda to COMS is

```
CREATE AGENDA <agenda name> OF <window name>
   PROCESSING_ITEM_LIST = <processing_item_list name>,
   DESTINATION = <program name>;
```

Updating Formats

To update formats, run the GEMCOS Utility against the TCL. Then disable the library by entering "?DISABLE LIBRARY <library name>". For more information about updating formats, refer to the COMS Operations Guide.

Amending Application Programs

Application programs must be amended to use the COMS direct interface. This means that for output formats the FORMID should be placed in the first six bytes of the COMS conversation area. If there is no conversation area, the application is terminated with an invalid-index error.

Obtaining Monitor Output

The Format Support Library sends format exception notification to the COMS monitor stations. This notification is directed to the window on the monitor station that corresponds to the window on which the exception occurred at the user station. Thus, if notification is desired, the monitor station must have the same window name (the GEMCOS system name) declared, and that window must be open.

Obtaining Monitor Output

The Format Support Library sends format exception notification to the COMS monitor stations. This notification is directed to the window on the monitor station that corresponds to the window on which the exception occurred at the user station. Thus, if notification is desired, the monitor station must have the same window name (the GEMCOS system name) declared, and that window must be open.

WHAT THE GEMCOS TCL FILE MUST CONTAIN

The GEMCOS TCL file must be compiled, without any syntax errors, with the GEMCOS/UTILITY, release 8.0 or higher. (A Series GEMCOS users need not recompile their TCL. The Format Support Library can be used with existing GEMCOS files as long as the files were generated by an 8.0-level GEMCOS/UTILITY.) If the release levels are incorrect at run time, an error is displayed and the Format Support Library goes to end of task (EOT) without being used.

The Format Support Library needs information from the device and format sections in the GEMCOS TCL file. In addition, it needs to know the system names defined in the TCL file so that the correct format can be applied according to which window the user is on. (This information is obtained from the files produced by the GEMCOS/UTILITY.)

The following files must be file-equated when the GEMCOS/UTILITY is run. The variable <codefile name> is the code file name of the Format Support Library. These files must have a security classification of PUBLIC I/O.

GEMCOS File	File-Equated to
IQUS	<codefile name>/INPUT
OQUS	<codefile name>/OUTPUT
CQUS	<codefile name>/CONTROL
FMTFILE	<codefile name>/FORMAT

The COMS Format Support Library provides that all the formats in a GEMCOS TCL file are made RESIDENT if PAGINGPERMANENT is assigned the value TRUE in the global section of the TCL file. If you have been running GEMCOS with PAGINGPERMANENT as TRUE in order to keep the GEMCOS PAGER stack in the mix, you should consider that PAGINGPERMANENT has a different function when running through COMS. You might want to assign the value FALSE to PAGINGPERMANENT if you do not want all formats to be RESIDENT.

4 GEMCOS WINDOWS

This section presents guidelines for running the Generalized Message Control System (GEMCOS) in a Communications Management System (COMS) Message Control System (MCS) window, and considers various aspects of GEMCOS operation through COMS. The section indicates limitations to GEMCOS features when GEMCOS is operating with the CP2000 front-end processor through COMS.

The information in this section pertains specifically to running GEMCOS in a COMS MCS window. For information about a COMS MCS window in general, refer to the "MCS Windows" section.

The following topics are presented in this section:

- General description of a GEMCOS window.
- Advantages of running GEMCOS through COMS.
- Changing station names in GEMCOS Transaction Control Language (TCL).
- Entering Control and Operator Display Terminal (ODT) commands.
- GEMCOS feature limitations with the CP2000.
- Relative GEMCOS functional behavior with the CP2000 and the Network Support Processor (NSP).

Unless indicated otherwise, the information in this section applies to GEMCOS operating either with the CP2000 through COMS or with the Network Support Processor (NSP) through COMS. Note that the information in this section under "GEMCOS Feature Limitations with the CP2000" applies only to GEMCOS operating with the CP2000 through COMS.

GENERAL DESCRIPTION OF A GEMCOS WINDOW

An MCS window that is used to run GEMCOS through COMS is called a GEMCOS window.

The Kernel version of COMS allows a station to have one dialog of the GEMCOS window.

The Entry version of COMS and the full-featured version of COMS provide a means to define a GEMCOS window that allows multiple simultaneous dialogs with GEMCOS at a station, once the GEMCOS system has been installed on the network. Such window definition is accomplished through the COMS Utility program, as described in the COMS Configuration Guide. A GEMCOS window can be defined to allow a maximum of eight simultaneous dialogs with the GEMCOS system at a station.

A number of GEMCOS windows can be defined through the COMS Utility. If the network is to include more than one GEMCOS system, a GEMCOS window can be defined and established for each of the systems.

ADVANTAGES OF RUNNING GEMCOS THROUGH COMS

A distinct advantage of running GEMCOS through COMS, rather than running GEMCOS directly under the NSP--or directly under the Data Communications Data Link Processor (DC-DLP)--is your ability to have multiple simultaneous dialogs with GEMCOS. This ability allows efficient use of a terminal.

Another advantage of running GEMCOS through COMS is the convenient menu-driven transaction-processing facility provided to COMS users by the Menu-Assisted Resource Control (MARC) program. For information about the MARC program, refer to the Menu-Assisted Resource Control (MARC) Operations Guide.

CHANGING STATION NAMES IN GEMCOS TCL

When GEMCOS is installed on the same network with COMS and window-defined through the COMS Utility program, the names of the stations in the TCL for the GEMCOS-controlled environment must be changed. The reason for changing the names of these stations is to make the names compatible with the COMS naming convention for pseudo-stations. In other words, the name of each station in the TCL must be made the same as the name of the corresponding COMS-allocated pseudo-station so that GEMCOS can identify which pseudo-stations it can communicate with and control.

For One Dialog

To be compatible with the COMS naming convention for pseudo-stations, you must change the Network Definition Language II (NDLII) name of each station in the TCL to reflect the following information and format:

<NDLII station name>/<MCS window name>/<dialog number>

This change typically involves adding a window name and dialog number to each of the station names that already exist in the TCL. For example, assume that the TCL includes a station named "STA1". If an MCS window named "GEMCOS" were declared for station "STA1" through the COMS Utility, it would be necessary to change the name of the station to "STA1/GEMCOS/1" in the TCL.

For More Than One Dialog

Note that the change in the example above only makes use of dialog 1. To use more dialogs, you duplicate the description of "STA1/GEMCOS/1" in the TCL as many times as required, and make the corresponding changes to the dialog number.

For example, to use three dialogs from the station, you duplicate the description of station "STA1/GEMCOS/1" in the TCL two times, and change the dialog number to 2 and 3. The result is three stations declared in the TCL (instead of the former single station "STA1"). You must also make all changes to device type and other areas in the TCL for the two new stations, to correspond to the way the original station was declared in the TCL. Consider that GEMCOS views each dialog as a different station, and makes no association between dialogs.

When GEMCOS is installed and window-defined and when the station names are changed in the TCL, you then can have a number of simultaneous dialogs with GEMCOS at your station.

ENTERING CONTROL AND ODT COMMANDS

The following information is provided to help a station user communicating through a GEMCOS window enter control commands directed to COMS or GEMCOS. Information is also provided to help an Operator Display Terminal (ODT) user who is communicating through a GEMCOS window enter system ODT commands.

When GEMCOS is operating with the CP2000 through COMS, datacomm network control for GEMCOS is handled by COMS instead of by GEMCOS. Therefore, refrain from using GEMCOS commands to manage the network; instead, use the corresponding COMS commands.

When GEMCOS is operating with the NSP through COMS, datacomm network control for GEMCOS is still handled by GEMCOS. Therefore, you can use GEMCOS commands to control parts of the network. Refer to "Requirements for an MCS to Function Through COMS" in the "MCS Windows" section for information.

See also
Requirements for an MCS to Function Through COMS. 16

You can enter GEMCOS commands not associated with network control at any time while you are communicating through a GEMCOS window, whether COMS is operating with the CP2000 or the NSP.

To direct a command to COMS from a GEMCOS window, enter the command prefixed by one question mark (?). To direct a command to GEMCOS from a GEMCOS window, enter the command prefixed by two question marks (??). To enter a system ODT command from a GEMCOS window, enter the command prefixed by three question marks (???).

Be sure to prefix your command with the correct number of question marks.

If you enter a command prefixed by one question mark (?), COMS attempts to handle the command, even if it is a GEMCOS command. If COMS understands the command, it handles it. If not, it passes the command to GEMCOS.

GEMCOS handles all commands prefixed by two question marks (??). Therefore, to ensure that GEMCOS handles a command, enter it prefixed by two question marks (??). For example, both COMS and GEMCOS have a STATUS command. If STATUS is prefixed by one question mark (?), COMS responds with the status. If STATUS is prefixed by two question marks (??), GEMCOS responds with the status.

Note that a command like CHANGE is always handled by GEMCOS, whether prefixed by one or two question marks. Since there is no CHANGE command in COMS, COMS gives the CHANGE command prefixed by one question mark (?) to GEMCOS.

GEMCOS FEATURE LIMITATIONS WITH THE CP2000

Operating GEMCOS with the CP2000 through COMS entails a number of limitations to GEMCOS features. These limitations are described in the following paragraphs.

Limitations Due to Loss of Toggles and Tallies

Toggles and tallies are the station-oriented variables in the NDLII program. When GEMCOS is operating either directly under the NSP or with the NSP through COMS, GEMCOS uses toggles and tallies to communicate with the NSP.

The CP2000 has built-in request sets and does not recognize toggles and tallies. Therefore, GEMCOS cannot use toggles and tallies when it is operating with the CP2000 through COMS.

The inability OF GEMCOS to use toggles and tallies when it is operating with the CP2000 through COMS results in the following limitations to GEMCOS features:

1. The STAMASTER attribute in the station section of the GEMCOS TCL is not supported. Since GEMCOS can no longer use the STAMASTER attribute to designate a station to be a master station, no computer connection that uses the STAMASTER attribute is supported.

2. If an object job is used, all stations to which the object job is attached must have MYUSE equal to 3 (that is, all stations must be input- and output-capable). GEMCOS is unable to inform the NSP that it is sending a message to a network-control station on object job output.

3. The BROADCAST attribute in the station section of the GEMCOS TCL is not supported. Therefore, GEMCOS cannot use the BROADCAST attribute to instruct the NSP to broadcast a message to all stations on a line. Note that a message can still be broadcast to all stations on a line, but the message is sent to each station one at a time rather than simultaneously to all stations.

4. When the MSG-QBYPASS bit (COMMON[7].[46:1]) is assigned the value 1 by a user program, the output message is not audited. It is recommended that the MSG-QBYPASS bit (COMMON[7].[46:1]) be equal to 0 (zero). If the bit is equal to 1 and a station gives negative acknowledgement (NAK) to an output message, the message is lost.

5. The NOQUEUE attribute in the station section of the GEMCOS TCL is not supported.

Other Limitations

When GEMCOS is operating with the CP2000 through COMS, limitations to GEMCOS features other than the limitations due to the loss of toggles and tallies are the following:

1. The NOLINE and IDENTIFY attributes in the station section of the GEMCOS TCL are not supported because COMS, and not GEMCOS, handles the dial-in stations. Therefore, you cannot use a dial-in station to dial in to GEMCOS.

2. The LOGICALACK attribute in the station section of the TCL in GEMCOS is not supported. Therefore, GEMCOS cannot use the NSP LOGICALACK (logical acknowledgement) function.

3. The SETMCSREQUESTSET attribute in the station section of the GEMCOS TCL is not supported. Therefore, GEMCOS cannot use the SETMCSREQUESTSET attribute to instruct the NSP to switch between request sets.

4. Process programs associated with GEMCOS are no longer able to issue network control commands to GEMCOS to control the datacomm network. In this respect, the process programs are limited in the same way that station operators are limited in the ability to control the network. All network control is handled by COMS.

RELATIVE GEMCOS FUNCTIONAL BEHAVIOR WITH CP2000 AND NSP

Because of the limitations described under "GEMCOS Feature Limitations with the CP2000", GEMCOS operating with the NSP through COMS is expected to perform more functions and provide more results than GEMCOS operating with the CP2000 through COMS. However, GEMCOS operating with the CP2000 through COMS is still expected to have most of the capabilities of GEMCOS operating with the NSP through COMS.

Further, GEMCOS operating with the NSP through COMS is expected to have almost all of the capabilities of GEMCOS operating directly under the NSP.

5 REMOTE FILES

This section compares the handling of remote files through the Communications Management System (COMS) with the handling of remote files through the Command AND Edit (CANDE) system. CANDE uses logical I/O to handle remote files, while COMS also has extensions for the window handling. Emphasis is placed in this section on how COMS handles remote files.

For a detailed discussion of remote files in a COMS environment, refer to the COMS Programming Guide and the COMS Configuration Guide.

WHAT IS A REMOTE FILE?

A remote file is a file with a KIND attribute specified as REMOTE. A remote file enables object programs to communicate interactively with a data communications station.

You typically use a remote file when you want to dedicate a program to a certain user. A remote-file program characteristically involves data entry or inquiry on records from a station.

SITUATIONS FOR USING REMOTE-FILE WINDOWS

In operating under COMS, there are basically two situations in which you would consider using remote-file windows:

1. If you have existing programs that use an existing remote-file interface.

2. If you want to dedicate a program to a certain user.

COMPARISON OF COMS AND CANDE REMOTE-FILE HANDLING

If you are operating under CANDE and execute a remote-file program, the remote-file program declares a remote file that by default works only with the single station that it recognizes. In other words, a remote-file program executed under CANDE can have only one user, unless it explicitly takes action to add more stations to its remote file. Separate copies of the program are needed for other users.

Under COMS, on the other hand, you can have remote files that accommodate a number of stations (multistation remote files) as well as single-station remote files.

When you run a remote-file program under CANDE, either in a CANDE window under COMS or through CANDE directly controlling your real station, the remote file takes over the station, and all input except control commands is routed to the remote file. This is also true if you run a remote-file program outside of CANDE--for example, through Work Flow Language (WFL)--provided the remote file is label-equated to your station.

When you run a remote-file program under COMS, either by running it in the Menu-Assisted Resource Control (MARC) interface or by running it outside of COMS/MARC with the remote file label-equated to your station, COMS creates for it a dynamic remote-file window called "REM<number>". When the program is run from MARC, you can "move" to the remote file either by specifying "TASK" as a command to MARC or by specifying "?ON REM<number>/<dialog number>". When the program is run from the outside, you have only the option "?ON REM<number>/<dialog number>". When the dynamic remote-file window is the current window, all input except control commands is routed to the remote file.

A program that runs in CANDE can typically run under COMS as a dynamic remote-file program without any change.

Note that operating a remote-file program from a station under CANDE in a COMS MCS window is no different from operating a remote-file program from a station controlled directly by CANDE.

Dynamic and Declared Remote Files

CANDE handles only dynamic remote files. COMS handles both dynamic remote files (that is, not defined to COMS) and declared remote files (that is, defined through the COMS Utility). Under COMS, dynamic remote-file programs are not shared. However, declared remote-file programs can be shared as defined through the COMS Utility.

A declared remote-file window is a window defined for an associated remote-file program through the COMS Utility. A declared remote-file program can handle either one station or a number of stations. The number is specified through the COMS Utility. Specific logic is normally required in the program to handle more than one station. A program that runs in CANDE can run under COMS as a declared remote-file program, but this declared program typically is not written to handle more than one station.

Note that if a remote-file program receives parameters or requires a run-time file equation or task equation, it cannot be used as a declared remote-file program under COMS. Moreover, remote-file programs that access attributes of the remote file might require modifications to work properly as declared remote-file programs.

Multistation Remote Files

Under CANDE, a remote-file program can access more than one station in a remote file by either of two means. It can add stations to the file and subtract stations from the file by using the STATIONLIST file attribute as described in the I/O Subsystem Programming Reference Manual, or it can open a file using a title of a file described in the installation Network Definition Language II (NDLII).

When COMS is used, a multistation remote-file program does not have to use either of the means indicated above to add and subtract stations. COMS does all the necessary adding and subtracting. When you first access the window (using "?ON <declared remote-file window>"), COMS adds the station to the file without help from the program. When you close the window (using "?CLOSE" or by logging off), COMS subtracts the station from the file. In this case, the remote-file program is informed of the closure by getting an end-of-file (EOF) indication for the remote station number (RSN) of the station.

Input Remote Files

Under CANDE, you can have only one input remote file. Under COMS, you can have more than one input remote file, each in a separate dynamic remote-file window.

Defining and Executing Declared Remote-File Programs

Because CANDE handles only dynamic remote files, you cannot define remote-file programs through CANDE.

Through the COMS Utility, you can define certain remote-file programs to COMS, including user-written programs that meet simple restrictions. You can then assign these programs unique window names, and make the windows available to all or selected users. These are declared remote-file windows. A user can then specify "?ON <name of declared remote-file window>)" and execute the program. For information on how declared remote-file windows are defined through the COMS Utility, refer to the COMS Configuration Guide.

Executing a Remote-File Program by a RUN Command

In a COMS environment, any dynamic remote-file program that you execute by a RUN command from a CANDE window or a MARC window opens a remote file to your station that shares the original window. No remote-file window is involved for programs executed in this manner.

Closing a Remote File

Under COMS, remote-file programs receive an end-of-file (EOF) indication on their remote files whenever the window associated with the program is closed. If the LASTSTATION file attribute is interrogated at this point, it specifies the remote station number (RSN) of the station removed from the remote file to indicate that CLOSE has occurred for that station. Declared remote-file programs receive an additional EOF indication when COMS instructs the program to go to the end of the job (EOJ). In this case, the LASTSTATION attribute is assigned the value "RSN 1". This occurs when all users have closed a remote-file window and the minimum number of copies for that program has been exceeded.

The following program fragment is an example of the statements that might be used in a multistation remote-file program written in ALGOL and running under COMS. The fragment shows how the program handles an EOF on its remote file when the window associated with the program is closed.

```
DEFINE
ERROR_FLAG = [0:1] #,
EOF_FLAG   = [9:1] #;

WHILE TRUE DO
BEGIN
READBOOL := READ(RMT, 288, IN_BUF);
IF NOT READBOOL.ERROR_FLAG THEN
PROCESS_INPUT(IN_BUF)
ELSE
IF READBOOL.EOF_FLAG THEN
IF RMT.LASTSTATION = 1 THEN
WRAP_UP_PROGRAM
ELSE
HANDLE_CLOSE_OF-STATION
ELSE
.
.
.
END;
```

The following program fragment is an example of the statements that might be used in a remote-file program written in COBOL and running under COMS. The fragment shows how the program handles an EOF on its remote file when the window associated with the program is closed.

```
77 LAST-STA-SW          PIC 9     VALUE 0.
88 LAST-STA                     VALUE 1.
.
.
MAINLINE.
.
.
.
PERFORM RECEIVE-INPUT THRU RECEIVE-INPUT-EXIT
UNTIL LAST-STA.
PERFORM WRAP-UP-PROGRAM.
STOP RUN.

RECEIVE-INPUT.
READ RMT;
AT END
PERFORM HANDLE-EOF
GO RECEIVE-INPUT-EXIT.
PERFORM PROCESS-INPUT.
RECEIVE-INPUT-EXIT.
EXIT.

HANDLE-EOF.
IF ATTRIBUTE LASTSTATION OF RMT = 1 THEN
MOVE 1 TO LAST-STA-SW
ELSE
PERFORM HANDLE-CLOSE-OF-STATION.
```

Tanking

Logical I/O, which is used by CANDE, does all the tanking for a remote file on a file basis. COMS, on the other hand, does all the tanking on a dialog basis for each station in a remote file. This means that COMS cannot apply the same rules for suspending output as logical I/O does. Logical I/O allows as many outstanding messages as there are file buffers for the file before suspending output. COMS allows 2400 characters per station, including datacomm headers, to be outstanding.

Handling of Time Limit on a Write Operation

Time-limit handling by logical I/O differs from time-limit handling by COMS. Logical I/O, which is used by CANDE, gives a time-limit exception when no file buffers are available for the write operation for the specified time period. Except as noted below, COMS gives a time-limit exception when the number of outstanding characters for a station has exceeded the limit of 2400 for the specified time period.

Note that if a broadcast write operation is done, COMS gives a time-limit exception only if no station in the remote file has been sent output.

Direct I/O

Direct I/O on remote files is supported by logical I/O only if the controlling MCS, typically CANDE, controls the Network Support Processor (NSP) stations directly. Direct I/O on remote files is not supported by COMS, nor is it supported in a COMS MCS window.

If a program that uses direct I/O on remote files is run in a COMS remote-file window, COMS handles the remote file as if it were a nondirect file with tanking set to ASYNC. In handling this I/O on the pseudo-station for the MCS window, the Master Control Program (MCP) completes the I/O when COMS gets the transaction, not when the transaction is sent to the terminal.

A GEMCOS-TO-COMS FUNCTION AND FEATURE COMPARISON

Appendix A consists of seven tables that compare the functions and features supplied by the Generalized Message Control System (GEMCOS) with those provided in the full-featured Communications Management System (COMS).

Each table compares one functional area. The tables are presented in alphabetical order:

- Table A-1. Application Program Interface
- Table A-2. Computer-to-Computer Communication
- Table A-3. Environment Interface
- Table A-4. Message Formatting
- Table A-5. Recovery
- Table A-6. Security
- Table A-7. Station Interface

Within each table, the features are listed alphabetically. Aspects of the feature are shown in alphabetical order.

Several GEMCOS features, while not supplied in COMS, can be implemented through a processing item or station list. When applicable, methods for implementing these features are noted in the tables.

Table A-1. Application Program Interface (Page 1 of 3)

GEMCOS Feature	COMS Feature
Batch job interface	No (Synchronized recovery with batch jobs is available outside COMS.)
Conversation areas	Yes (Available on input to the program from a processing item.)
Editor program	No (Can be implemented by a user-created processing item.)
Module/function indexes	Yes
Multiple input queues per program	No
Multiple transactions per execution	Yes
Object job interface	
Program	No
Station	Yes
Practice mode	No
Priority scheme for input queues	No
Program initiated by NOINPUT command	No
Program can specify TITLE on output to indicate message sent to multiple stations	No (Can be implemented by a user-created processing item or a station list.)

GEMCOS-to-COMS Function and Feature Comparison

Table A-1. Application Program Interface (Page 2 of 3)

GEMCOS Feature	COMS Feature
Retry counter for Transaction Processor (TP) failure	Yes
Sending multiple output messages per transaction	Yes
Sending/receiving parts of messages	No (Can be implemented by a user-created processing item or a station list.)
Service program	No (Can be implemented by a user-created processing item or a station list.)
Single transactions per execution	No
Station bits	Yes (COMS Installation data feature provides the same functionality.)
Test Program	No
Transaction-based routing as follows:	
Program-to-program response to program	No (Can be implemented by a user-created processing item.)
Program-to-program response to station	Yes
Station to program	Yes
User-supplied routing	No (Can be implemented by a user-created processing item.)

Table A-1. Application Program Interface (Page 3 of 3)

GEMCOS Feature	COMS Feature
Transmission of messages to	
Area broadcast	No (Can be implemented by a user-created processing item or a station list.)
Area rotary	No
Monitor station	No (Can be implemented by a user-created processing item or a station list.)
Network control station	No (Can be implemented by a user-created processing item or a station list.)
Originator	Yes
Program by number	Yes (Sent by program agenda.)
Program by trancode	Yes
Station by name	No (Can be implemented by a user-created processing item or a station list.)
Station by number	No (Can be implemented by a user-created processing item or a station list.)
Stations by output message ID	No (Can be implemented by a user-created processing item or a station list.)
Validation of area name	No
Validation of station name	Yes

GEMCOS-to-COMS Function and Feature Comparison

Table A-2. Computer-to-Computer Communication

GEMCOS Feature	COMS Feature
GEMCOS-to-GEMCOS file transfer	No
Routing headers for	
Station GEMCOS-to-GEMCOS communication of transactions and output messages	No
Station as port file	No
Station that is port file	No
Transaction-based routing station to station by trancode	No

Table A-3. Environment Interface (Page 1 of 2)

GEMCOS Feature	COMS Feature
Administrative message switching	No
COMPUTE command	No
Control of	
Data base programs	Yes
Specified stations	Yes
Dynamic volume	Yes
Interception of station message traffic	No (Can be implemented by a user-created processing item.)
Limitation of memory for program input messages	Yes
Memory subsystem control for	
Message Control System (MCS)	Yes (Only for data base control.)
Programs	Yes

GEMCOS-to-COMS Function and Feature Comparison

Table A-3. Environment Interface (Page 2 of 2)

GEMCOS Feature	COMS Feature
Output Message	
Refresh	No (Can be implemented by a user-created processing item.)
Recall	No (Can be implemented by a user-created processing item.)
Placing station out of service for 15 minutes	No
Program in mix can be	
Permanent in mix	Yes
Temporary in mix	Yes
Restricting commands from user-supplied program	Yes
Station alternates	No (Can be implemented by a user-created processing item.)
Statistics	Yes

Table A-4. Message Formatting (Page 1 of 2)

GEMCOS Feature	COMS Format Support Library Feature
Blank fill	Yes
Data translation [FUNCTION]	Yes
Forms composition	Yes
Input formatting by trancode	Yes
Item count passed to program	Yes
Numeric editing	Yes
Numeric verification and zero fill	Yes
On-line updating of formats	Yes
Output formatting by device type and output message ID	Yes

GEMCOS-to-COMS Function and Feature Comparison

Table A-4. Message Formatting (Page 2 of 2)

GEMCOS Feature	COMS Format Support Library Feature
Paging	
Input	Yes
Output	Yes
Paging breaks	Yes
Repeat paging	Yes
Total lines	Yes
Update	Yes
Variable format	Yes
Updated formats available in practice mode	No
User formatting of MCS responses	No (Can be implemented by a user-created processing item. Note that all COMS Message Control System (MCS) messages go through the Menu-Assisted Resource Control interface (MARC), and MARC has multilingual system (MLS) capabilities to translate the language.)
User-supplied formatting routines	Yes
Variable length of items	Yes
Variable repetition of items	Yes
Variable sequence of items	Yes

Table A-5. Recovery

GEMCOS Feature	COMS Feature
Archival recovery	Yes
Audit of input messages	Yes
Audit of input messages by trancode	Yes (Auditing done by data base.)
Audit of output messages	No (Can be implemented by a user-created processing item.)
Audit of output messages by program action	No (Can be implemented by a user-created processing item.)
DMSII	
Recovery of program-to-program response to program messages	No
Recovery of program-to-program response to station messages	No
Synchronized recovery	Yes
Nonrecovery of messages	Yes
Recovery of output message before entering Data Management System II (DMSII) transaction state	No
Recovery for batch job feeding transaction programs	Yes
Recovery transparent to program	Yes

GEMCOS-to-COMS Function and Feature Comparison

Table A-6. Security

GEMCOS Feature	COMS Feature
Continuous log-on	Yes
Dial-in stations required to identify	No
Identifier passed to programs [INDIVIDUALID]	Yes
Multiple users per station	No
Station able to switch between application systems	Yes
Station limited to trancodes for application system	Yes (Usercode limited to windows.)
Usercode (no password)	Yes (Usercode with password.)
User-supplied access control package	No (Can be implemented by a user-created processing item.)
Valid trancodes for usercode	Yes
Valid usercodes for station	Yes

Table A-7. Station Interface

GEMCOS Feature	COMS Feature
Acknowledgement of input messages as follows:	
LOGICALACK by MCS	Yes (Not on CP2000 stations.)
PROGRAMACK by program	No
Fixed length of trancodes	Yes
Limit of one transaction at a time	Yes
Return to Network Definition Language II (NDLII)-defined Message Control System (MCS) on end of job (EOJ)	Yes
Select tailored datacomm software	No
Specify location of trancode in message	Yes
Title on output messages	No (Can be implemented by a user-created processing item.)

B GEMCOS SOURCES FOR COMS ENTITIES AND ATTRIBUTES

An entity is a category of information. Attributes describe the entity or are components of the entity. For Generalized Message Control System (GEMCOS) programs to run directly as Communications Management System (COMS) programs, GEMCOS entities must be converted into COMS entities.

The following list shows the GEMCOS counterpart for six COMS entities: agendas, programs, trancodes, device types, stations, and windows. The entities are listed as they appear in the COMS Utility Home Menu, where COMS entities are normally defined. The list shows all attributes that can be defined for that entity using the COMS Utility. If the attribute does not have a GEMCOS counterpart, "---" appears in the GEMCOS column.

Agenda Counterparts

GEMCOS Entity	COMS Entity
INPUTQUEUE <ID> + AG	Agenda name
SYSTEM <ID>	Window name
---	Default agenda
---	Processing-item list
PROGRAM <ID>	Destination

Program Counterparts

To alleviate migration problems and to identify the COMS program entity correctly, do the following:

1. Define the USERCODE attribute if the TITLE attribute has a specified usercode.

2. In the FAMILY attribute, include the family on which COMS resides.

GEMCOS Entity	COMS Entity
PROGRAM <ID>	Program name
PROGRAM TITLE	TITLE
PROGRAM USERCODE	USERCODE
PROGRAM FAMILY	FAMILY
PROGRAM CHARGECODE	CHARGECODE
---	SUBSYSTEM
PROGRAM PRIORITY	PRIORITY
PROGRAM SUBSPACES	SUBSPACES
---	Valid security-category list
---	Database Name
---	Input-queue memory size
PROGRAM MINCOPIES	Minimum copies
PROGRAM MAXCOPIES	Maximum copies
---	Remote-file interface
1st INPUTQUEUE TIMELIMIT	Time limit
1st INPUTQUEUE QUEUEDEPTH	Queue depth
---	Remote users

GEMCOS Source for COMS Entities and Attributes

Trancode Counterparts

GEMCOS Entity	COMS Entity
MKE <ID> OF INPUTQUEUE	Trancode name
SYSTEM <ID>	Window name
INPUTQUEUE <ID>AG	Agenda name
---	Security-category name
MKE <ID>[<module><function>]	Module Function Index

Device Type Counterparts

GEMCOS	COMS
DEVICE <ID>	Device-type name or names

Station Counterparts

GEMCOS	COMS
STATION <ID>	Station name
SYSTEM <ID>	Default window
DEVICE <ID> or STALIST	Device type
---	Default usercode
---	Valid security-category list
STATION MKEPOSITION	Trancode position
STATION SYSTEMPOSTATION GLOBAL SYSTEMSPO	Control station
---	Super user
---	System user
---	Privileged user
STATION CONTINUOUSLOGON	Continuous log-on

Window Counterparts

GEMCOS Entity	COMS Entity
SYSTEM <ID>	Window name
---	Maximum users
---	Maximum dialogs
17	Maximum trancode size
---	Remote-file window
---	Remote-file program
---	Message Control System (MCS) window
---	MCS title
---	"Notify Open" activity
---	Notification text
---	"Notify On" activity
---	Notification text

GLOSSARY

agenda

A COMS entity used for message routing. An agenda consists of a processing-item list and, if needed, a destination. You can apply an agenda to messages received and sent by application programs.

Agenda Processor library

A COMS internal library that applies processing items to messages by executing the processing-item lists that agendas specify.

attribute

Information that describes the characteristics of an entity.

BNA

See "Burroughs Network Architecture."

Burroughs Network Architecture (BNA)

A network architecture used on Unisys computer systems to connect multiple, independent, compatible Unisys computer systems into a network for distributed processing and resource sharing.

CANDE

See "Command AND Edit."

Command AND Edit (CANDE)

A Message Control System (MCS) that provides generalized file preparation and updating capabilities and task control in an interactive, terminal-oriented environment.

Communications Management System (COMS)

A general Message Control System (MCS) that controls on-line environments on the A Series and B 5900, B 6900, and B 7900 systems. COMS can support the processing of multiprogram transactions, single-station remote files, and multistation remote files.

COMS

See "Communications Management System."

COMS Utility

A COMS program that defines and maintains the specifications stored in the COMS configuration file.

conversation area

The user data space in the header of a COMS message. The conversation area is user defined. It can contain information passed by a program or processing item.

Data Communications Controller (DCC)

The subset of the Master Control Program (MCP), operating as a group of independent tasks, each associated with one Network Support Processor (NSP), that is the basic interface between the data communications subsystem and the main system.

Data Communications Data Link Processor (DC-DLP)

A data communications processor that combines the functions of the Network Support Processor (NSP) and Line Support Processor (LSP) into one physical Data Link Processor (DLP) and supports up to four lines of communication.

Data Management System II (DMSII)

A specialized system software package used to describe a data base and maintain the relationships between the data elements in the data base.

DCALGOL

Data Communications ALGOL. A language containing ALGOL extensions allowing it to be used to write Message Control Systems (MCSs) and other specialized system programs.

DCC

See "Data Communications Controller."

DC-DLP

See "Data Communications Data Link Processor."

direct window

A COMS window that allows you to route messages directly to COMS while using all the COMS capabilities for preprocessing and postprocessing of messages.

DMSII

See "Data Management System II."

end of file (EOF)

A code at the end of a data file that signals that the last record in the file has been processed.

end of job (EOJ)

The termination of processing of a job.

end of task (EOT)

The termination of processing of a task.

entity

A category of information within a COMS configuration file or within a Generalized Message Control System (GEMCOS) control file.

entry point

A procedure in a library program that can be called by another program.

EOF

See "end of file."

EOJ

See "End of Job."

EOT

See "end of task."

Format Support Library (FS-LIB)

A COMS-provided library used to maintain Generalized Message Control System (GEMCOS) input formatting, output formatting, forms request, paging, and library formats.

formatting

The handling of fields within a message prior to their delivery to a program (input message) or a terminal (output message).

FS-LIB

See "Format Support Library."

GEMCOS

See "Generalized Message Control System."

Generalized Message Control System (GEMCOS)

A Message Control System (MCS) developed for on-line, multitask Message-Level Interface Processor (MLIP) systems. GEMCOS is transaction oriented.

input format

A format for messages to be delivered to a program.

library

A collection of one or more named routines or entry points that are stored in a file and can be called by other programs.

MARC

See "Menu-Assisted Resource Control."

MCS

See "Message Control System."

Menu-Assisted Resource Control (MARC)

A menu-driven interface and transaction processor for users of A Series and B 5900, B 6900, and B 7900 Series systems.

Message Control System (MCS)

A program that controls the flow of messages between terminals, application programs, and the Master Control Program (MCP). MCS functions can include message routing, access control, audit and recovery, system management, and message formatting.

MLS

See "MultiLingual System."

MultiLingual System (MLS)

A system for developing and accessing output messages, on-line help text, and menu screens in different natural languages, such as English, French, and Spanish.

NDLII

See "Network Definition Language II."

Network Definition Language II (NDLII)

A Unisys language used to describe the physical, logical, and functional characteristics of the data communications subsystem on Message-Level Interface Processor (MLIP) and Host Data Unit (HDU) systems to Network Support Processors (NSPs) and Line Support Processors (LSPs).

Network Support Processor (NSP)

A data communications subsystem processor that controls the interface between a host system and the data communications peripherals. It executes the code generated by the Network Definition Language II (NDLII) compiler for line control and editor procedures.

NSP

See "Network Support Processor."

output format

A format for messages to be delivered to a terminal.

paged format

A format for a message that cannot be displayed on a terminal in one screen, or for a message that is specifically divided into several screens.

postprocessing

The processing done to a message by processing items after an application program sends the message out.

preprocessing

The processing that the Agenda Processor does to a message before an application program receives the message.

processing-item library

In COMS, a user-written ALGOL library containing a set of procedures called processing items. The library can be called only by the Agenda Processor library of COMS to preprocess and postprocess messages.

remote file

A file with a KIND attribute specified as REMOTE. A remote file enables object programs to communicate interactively with a data communications station.

Remote Print System (ReprintS)

A Unisys software system that controls the routing and printing of backup files at remote (datacomm) destinations.

ReprintS

See "Remote Print System."

restart data set

In Data Management System II (DMSII), a data set containing restart records that application programs can use to recover data base information after a system failure.

Screen Design Facility (SDF)

The InterPro (tm) product used for creating screens for on-line, transaction-based application systems.

SDF

See "Screen Design Facility".

synchronized recovery

A COMS function that resubmits incomplete transactions to the data base after a transaction-state abort, system crash, or rollback. The synchronized recovery function processes transactions in the same order in which they were originally processed.

TCL

See "Transaction Control Language."

trancode

See "transaction code."

transaction code

In COMS, a code that can appear in a transaction-initiating message header indicating the processing that is to be carried out. This code is used to route the message to the appropriate host program.

Transaction Control Language (TCL)

In the Generalized Message Control System (GEMCOS), a language that is used to describe stations, programs, and Message Control System (MCS) capabilities.

transaction state

In Data Management System II (DMSII), the period in a user-language program between a BEGINTRANSACTION and an ENDTRANSACTION statement.

Glossary

window

A COMS architectural concept that enables you to operate a number of program environments independently and simultaneously at one station.

BIBLIOGRAPHY

The following documents are either referenced in this guide or are additional documents that may be useful to personnel responsible for their site's migration to COMS.

ALGOL Programming Reference Manual Volume 1: Basic Implementation (form 1169844). Unisys Corporation. Formerly ALGOL Reference Manual.

ALGOL Programming Reference Manual, Volume 2: Product Interfaces (form 1170099). Unisys Corporation.

B 7000/B 6000 Series Generalized Message Control System (GEMCOS) Capabilities Manual (form 1100211). Unisys Corporation.

B 7000/B 6000 Series Generalized Message Control System (GEMCOS) Formatting Guide (form 1100344). Unisys Corporation.

COBOL ANSI-74 Programming Reference Manual, Volume 1: Basic Implementation (form 1169877). Unisys Corporation. Formerly COBOL ANSI-74 Reference Manual.

COBOL ANSI-74 Programming Reference Manual, Volume 2: Product Interfaces (form 1180619). Unisys Corporation.

Communications Management System (COMS) Capabilities Manual (form 1180494). Unisys Corporation.

Communications Management System (COMS) Configuration Guide (form 1185238). Unisys Corporation. Formerly Communications Management System (COMS) Planning and Installation Manual.

Communications Management System (COMS) Operations Guide (form 1154523). Unisys Corporation. Formerly Communications Management System (COMS) Operator's Guide.

Communications Management System (COMS) Programming Guide (form 1185303). Unisys Corporation. Formerly Communications Management System (COMS) Programmer's Guide.

DCALGOL Programming Reference Manual (form 5014574). Unisys Corporation. Formerly DCALGOL Reference Manual.

Generalized Message Control System (GEMCOS) Operations Guide (form 1185329). Unisys Corporation. Formerly Generalized Message Control System (GEMCOS) User's Guide.

I/O Subsystem Programming Reference Manual (form 1169984). Unisys Corporation. Formerly I/O Subsystem Reference Manual.

Menu-Assisted Resource Control (MARC) Operations Guide (form 1169588). Unisys Corporation. Formerly Menu-Assisted Resource Control (MARC) User's Guide.

Operator Display Terminal (ODT) System Commands Operations Reference Manual (form 1169612). Unisys Corporation. Formerly Operator Display Terminal (ODT) Reference Manual.

Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide (form 1169919). Unisys Corporation. Formerly Print System (PrintS/ReprintS) User's Guide.

System Software Support Reference Manual (form 1170016). Unisys Corporation.

Agendas
   defining to COMS, 42
   in Format Support Library, 39
   input and output processing, 41
Application program
   amending to use COMS direct interface, 42

CANDE, See Command AND Edit (CANDE)
COBOL68, 6
COBOL74, 6
Command AND Edit (CANDE)
   operating under MCS window, 3, 8
   remote-file handling
      compared to COMS remote-file handling, 55
Communications Management System (COMS), 1
   CANDE window, 3
   coexisting with SDF, 33
   dial-out, 5
   direct migration, 5
   entities, 6
      compared with GEMCOS entities, 6
      required for Format Support Library, 39
   features, 1
   GEMCOS features supported under COMS, 7, 61
   GEMCOS windows in, See GEMCOS windows
   MARC window, 3
   MCS window in, See MCS window
   versions of COMS
      Entry version, 8, 13
      full-featured version, 1, 8, 13
      Kernel version, 8, 13
   windows
      defined at system initialization, 2
      MCS, 8, See also MCS window
      types of, 2
COMS, See Communications Management System (COMS)
COMS Utility
   defining the Format Support Library, 31
   use of, 2
Control commands, entering, 17
Control paging, 37
Conversation area
   importance to Format Support Library, 40
   SDF and the Format Support Library, 33
CP2000
   limitations to GEMCOS features with, 52
   operating under GEMCOS window, 10

DCWRITEs, 11, 18
   error messages for, 25
   fields handled by COMS, 27

DCWRITEs (cont.)
fields in result messages, 30
formats of result messages, 29
handled by the MCP, 18
interpreted by COMS, 19
message formats, 26
not allowed by COMS, 24
result messages given by COMS, 23
Devices
defining to COMS, 40
in Format Support Library, 39
Dial-out, 5
Direct I/O, 60
Direct migration, 5
Direct window, 2

Entities
COMS, 6
GEMCOS, 6
Entry version of COMS, 8, 13

Format library, 35
Format Support Library, 31
ACTUALNAME, 41
as direct migration aid, 5
defining COMS entities, 39
formatting under COMS, 7
in agendas, 36
logical flow
for forms request, 37
for input and output formatting, 36
for paging, 37
migration considerations, 32
order in agenda, 41
performance, 36
preparing GEMCOS TCL file, 43
required COMS entities, 39
requirements, 36
use with SDF, 33
Format updating, 42
Formatting
GEMCOS TCL file, 32
through Format Support Library, 31
Forms request, 34
logical flow for, 37
Full-featured version of COMS, 1, 8, 13

GEMCOS, See Generalized Message Control System (GEMCOS)
GEMCOS windows, 3, 45
advantages of running GEMCOS through COMS, 47

GEMCOS windows (cont.)
  changing station names in TCL, 48
  entering
    control commands, 50
    Operator Display Terminal (ODT) commands, 50
  GEMCOS feature limitations with the CP2000
    due to loss of toggles and tallies, 52
    other limitations, 53
  operating under COMS, 8
GEMCOS/FORMATLIBRARY, 35
Generalized Message Control System (GEMCOS)
  counterparts for COMS entities and attributes
    agendas, 73
    device types, 75
    programs, 74
    stations, 75
    trancodes, 75
    windows, 76
  entities, 6
  features supported under COMS, 7
  format library, 35
  formatting, 34
  forms request, 34
  installation for direct migration, 5
  operating under GEMCOS window, 8
  paging dialogue, 34
  paging messages, 34
  TCL file of, See TCL file
  windows, See GEMCOS windows

InterPro (tm)
  COMS, 1
  SDF, 7, 33

Kernel version of COMS, 8, 13

Libraries
  in Format Support Library, 39
Logical I/O
  for tanking, 60
  for time-limit handling, 60

MARC, See Menu-Assisted Resource Control (MARC)
MCS, See Message Control System (MCS)
MCS window, 2, 13
  as used for any given MCS, 11
  DCWRITEs handling, 18, See also DCWRITEs
  defining windows, 8
  dialogs, 13

MCS window (cont.)
established through pseudo-station feature, 14
maximum number allowed at a station, 13
entering control commands, 17
Menu-Assisted Resource Control (MARC)
installation for direct migration, 5
MARC window, 3
Message Control System (MCS)
requirements for functioning through COMS, 16
Migration
assistance, 10
benefits of COMS, 1
direct, 5
use of windows, 5, See also Windows

Paging
dialogue, 34
logical flow for, 37
messages, 34
Processing items
defining to COMS, 41
Processing-item libraries, 40
Processing-item lists
creating, 41
in Format Support Library, 39
use in agendas, 41
Pseudo-station
description of, 14
naming convention for, 15

Remote file, 55
closing under COMS, 58
declared, 56
defining and executing, 57
dynamic, 56
handling
through CANDE, 55
through COMS, 55
input remote file, 57
multistation, 57
program execution
by RUN command, 58
of declared remote-file program, 57
tanking
by COMS, 60
by logical I/O, 60
time limit on write operation
in COMS, 60
in logical I/O, 60
Remote printing, 5

Remote-file program, 55
   declared, 57
   dynamic, 58
Remote-file window, 2, 3
   declared, 56, 57
   dynamic, 56
   situations for using, 55
ReprintS, 5
Restart data set, 5

Screen Design Facility (SDF)
   formatting for ET 1100-compatible devices, 7
   use with Format Support Library, 33, 41
SDF, See Screen Design Facility (SDF)
Steps to direct migration, 5
Synchronized recovery, 2

TCL, See Transaction Control Language (TCL)
TCL file, 43
   compiling for Format Support Library, 43
   establishing message formats, 31
TCL_FORMATTER, 31
Trancodes
   as feature of COMS, 1
   defining to COMS, 40
   in Format Support Library, 39
Transaction Control Language (TCL), 31

User-created message formats, 35

Windows
   as feature of COMS, 1
   creating, 39
   GEMCOS, See GEMCOS windows
   in Format Support Library, 39
   operating under, 8
   types of, 2